SYSTEM FOR TRANSMITTING LOCAL AREA NETWORK (LAN) DATA FRAMES

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates in general to the transmission of data frames between local area networks (LANs) and in particular to a system for transmitting LAN data frames through an asynchronous transfer mode (ATM) crossbar switch.

2. Description of the Related Art:

Local area networks (LANs), such as ethernet or token-ring networks, are generally coupled via hubs. A hub is a system made of LAN adapters that communicate via a switch card. This switch card is implemented either as a parallel bus or a passive switch card, which further includes a matrix for achieving the connection between selected inputs and outputs.

Today, asynchronous transfer mode (ATM) technology is improving at a rapid rate. Most research developments in this field are concentrated in high-speed ATM networks instead of LANs. Extremely high speed ATM switches are now readily available and are utilized for transferring data between LANs coupled to the ATM switch. The utilization of the ATM technology for switching LAN frames requires a transformation of each LAN frame by splitting the LAN frame into ATM frames. This is accomplished by encapsulating each LAN frame in the ATM adaptation layer (AAL) format.

The LAN frame is transformed into ATM data packets in the AAL format via a special module. The LAN frame is then transferred to the switch card for switching.

Such a requirement results in two major drawbacks. Since the frame is converted into ATM cells, a header in each cell including protocol information (e.g., destination address) is required. A second drawback is that the transformation of the LAN frame into ATM cells and the encapsulation in the AAL format requires important and costly hardware and software.

When a first LAN adapter requests transmission of a LAN frame to a second LAN adapter, the first LAN adapter sends a requests to the ATM switch and a scheduler determines whether or not satisfying this request is possible, while considering an algorithm that determines the best data coupling to establish at each time. Accordingly, the grant signal to this request may be delayed before being returned to the requesting LAN adapter, while other requests are still waiting to be processed in the second LAN adapter.

Consequently, it would be desirable for a system for exchanging data between multiple LANs without delaying requests between LANs.

SUMMARY OF THE INVENTION

To overcome the foregoing and additional limitations in the prior art, the present invention provides an improved data transmission system including multiple local area networks (LANs) coupled by a hub that further includes multiple LAN adapters coupled to the LANs. The present invention further includes an asynchronous transfer mode (ATM) crossbar switch coupling all LAN adapters.

At least one of the LANs requests transmission of LAN data frames to several destination LANs. The LAN data frames are converted into concatenated slots of an identical size and transmitted via the ATM crossbar switch. Each of these requests are associated with a destination LAN, and the ATM crossbar switch includes a scheduler for transmitting grant signals (GNT) associated respectively with the requests. This enables the requesting LAN adapter to transmit LAN data frames to the destination LANs. The grant signals are transmitted in an order that depends on a predetermined algorithm controlling the scheduler regardless of the order the requests are transmitted by the requesting LAN adapter.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 illustrates a block diagram of an exemplary data transmission system including four local area networks (LANs) coupled by a hub according to a preferred embodiment of the present invention;

Figure 2 depicts a block diagram of an asynchronous transfer mode (ATM) crossbar switch utilized within the hub according to a preferred embodiment of the present invention;

Figure 3 illustrates a block diagram of the control logic of a scheduler according to a preferred embodiment of the present invention;

Figure 4A and **Figure 4B** depict diagrams illustrating the signals exchanged between the ATM crossbar switch depicted in **Figure 2** and the LAN adapters according to a preferred embodiment of the present invention;

Figure 5 illustrates a block diagram of a LAN adapter within the data transmission system according to a preferred embodiment of the present invention;

Figure 6 depicts a block diagram of the control logic of the ATM crossbar switch according to a preferred embodiment of the present invention; and

Figure 7 illustrates a diagram depicting the main signals exchanged in a LAN adapter according to a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference to the figures, and in particular, with reference to **Figure 1**, there is depicted a block diagram of a preferred embodiment of the present invention. Multiple local area networks (LAN) **10**, **12**, **14** and **16** are coupled by a hub **15**, which includes an ATM crossbar switch **18** and multiple LAN adapters **20**, **22**, **24** and **26**. LAN **10** is coupled to ATM crossbar switch **18** via LAN adapter **20**, LAN **12** is coupled to switch **18** via LAN adapter **22**, LAN **14** is coupled to ATM crossbar switch **18** via LAN adapter **24**, and LAN **16** is coupled to switch **18** via LAN adapter **26**.

Referring to **Figure 2**, an ATM crossbar switch utilized in the invention includes a data switch module **30**, a scheduler **32**, multiple LAN adapter connectors **34** and **36** coupling multiple LANs to ATM crossbar switch **18**, and a clock generator **38** for supplying the clock and the synchronization to data switch module **30**, scheduler **32**, and to LAN adapter connectors **34** and **36**.

Data switch module **30** includes a switching data block **40**, which is generally implemented as a passive switching matrix between data input signals from the LAN adapters to the switching matrix and data output signals from the switching matrix to the LAN adapters. Data switch module **30** also includes a control logic **42**, which decodes the configuration signals received from scheduler **32** to determine the data path connections and establishes the data path connection based on the synchronization signal received from clock generator **38**.

Scheduler **32** also includes an algorithm unit **46** which determines the best data connection to establish each time a request is issued by a requesting LAN. Such a determination is based on the selection of the request amongst all requests received from the requesting LAN adapters that meets some predetermined criteria such as a

priority order, the selection of unicast/multicast, the selection between reserved bandwidth data and non-reserved bandwidth data or any other criteria defined by the user.

A request signal (REQ) is issued by a requesting LAN adapter when the requesting LAN adapter has a LAN frame to transmit to a destination LAN adapter through the ATM crossbar switch. Such a request signal is a serially encoded signal during the first four bytes of a time slot and includes thirty-two bits, which are sampled by a signal sixteen times the data clock frequency. The first two bytes of the REQ signal includes the routing destination address on sixteen bits, one bit per LAN adapter, a bit being set when the destination address corresponds to the associated LAN adapter. This encoding scheme allows either a point-to-point connection, a multicast connection or a broadcast connection. The next two bytes of the REQ signal contains the connection time on sixteen bits, that is the number of time slots required to transmit the entire frame.

When receiving the grant signal (GNT) from control logic **44** of the scheduler, the requesting LAN adapter transmits its frame. The GNT signal generated by the control logic is a serially encoded signal during the first four bytes of a time slot and includes thirty-two bits, which are sampled by a signal sixteen times the data clock frequency. The first two bytes of the GNT signal includes the routing destination address on sixteen bits, one bit per LAN adapter, a bit being set when the destination address corresponds to a destination LAN adapter. As for the REQ signal, the next two bytes of the GNT signal includes the connection time on sixteen bits, or the number of slots utilized for transmitting the frame.

The GNT signal includes information that is identical to the information contained in the REQ signal because the GNT signal is entirely de-correlated from the

corresponding REQ signal. This means that any LAN adapter can both send multiple requests and receive multiple de-correlated signals.

With reference to **Figure 3**, there is illustrated a control logic **46** of scheduler **32** that includes functional units, which enable the requests from each adapter to be handled without delaying the requests due to a previous request that has yet to be processed. Control logic **46** includes a finite state machine FSM **50**, which controls all steps performed from the reception of a request signal REQ to the transmission of a grant signal GNT. Note that the FSM **50** is timed by a clock control unit **52**, which receives a clock signal and a SYNCHRO signal from the clock generator **38**, as depicted in **Figure 2**.

When a REQ is received from a LAN adapter, the REQ is deserialized by a serialized/deserializer SERDES **54**, and transferred under control of FSM **50** to a memory interface **56**. This REQ is stored into memory **58**. Note that multiple REQs can be stored into memory **58** for each LAN adapter.

Periodically or as soon as a REQ is received from a LAN adapter, memory interface **56** provides algorithm **46** with the REQs stored into memory **58** for each LAN adapter. Algorithm unit **46** selects a REQ among multiple REQs for each LAN adapter and provides REQ confirmation to FSM **50**. Such a configuration is sent by FSM **50** to a configuration control block (CFG) **60** for transmission to the data switch module **30**, as shown in **Figure 2**, on the configuration data lines. At the same time, the selected REQs are transferred to serializer/deserializer **54** to be serialized and transmitted as grant signals GNT to the LAN adapters **34** and **36**. For this, the frequency of clock signals issued from clock control unit **52** are multiplied by sixteen in multiplier **53** and the resulting signals are used by SERDES **54** for generating the thirty-two bits of the GNT signal. Note that FSM **50** also sends information signals to

memory interface **56** to indicate which REQs have been selected by algorithm unit **46** and have to be removed from memory **58**.

Referring now to **Figure 4A** and **Figure 4B**, the relationship between the signals at the interface between the ATM crossbar switch and LAN adapters is illustrated. First, the data clock pulses exchange the LAN frames between the adapters through the switch card. At least fifty-three clock pulses determine the time slot to exchange fifty-three data bytes, which correspond to the ATM cell size. It must be noted that there is no requirement on the clock rate.

The SYNCHRO signal is a one pulse clock during the first data byte of each time slot. The REQ signal is active during the first two data bytes of a time slot. Then, the algorithm process occurs during the following fifty-one bytes of the slot. Finally, a GNT signal delivered by the scheduler at the next time slot, is activated during the first two bytes of the following time slot. As previously mentioned the GNT signal corresponds to the selected configuration based on the algorithm processing.

Thus, as depicted in **Figure 4A** and **4B**, a request **X** for transmitting n slots is received at the beginning of slot **1**. At the same time a GNT signal is activated for a request **C**. Then, the request **X** is accepted by a GNT signal at the beginning of slot **2** during which data **C** are transmitted by the LAN adapter. At the beginning of the last slot (n+2) of the connection time during which the data **X** are transmitted by the LAN adapter, a REQ **Y** for two slots is accepted by a GNT signal. Then, the data corresponding to this GNT are transmitted by the LAN adapter during the two following slots, that is slots n+3 and n+4.

To summarize, at each synchronization pulse, control logic **44** stores all REQ configurations from each LAN adapter at each time slot. Then, algorithm **46**

determines the best connection possible based on all stored REQ, sets the configuration data lines for switch module **30** and activates the GNT signal to the selected LAN adapters. This new matrix switching state is latched into switching matrix **40** on the falling edge of the GNT signal. This is performed by control logic **42** of switching module **30**.

With reference to **Figure 5**, the hardware architecture of a LAN adapter includes a LAN logic **70** for processing the exchange of data with the LAN, a general bus **72** for transferring data bytes, a switch logic **74** for processing the exchange of data with the switch card, a system bus logic **76** for processing the transfer of data in the LAN adapter and an arbiter **78** for taking care of any bus contention for the requests which may come from LAN controller **74** or serial communication controller (SCC) **88**, as illustrated.

LAN logic **70** includes a LAN connector **80** that couples the LAN adapter to the LAN through a LAN attachment cable and carries the transmit data signal (TD) and the receive data signal (RD), an analog circuitry **82** for converting the TTL logic signals into analog or analog to TTL, and for providing specified network characteristics (e.g., impedance, capacitance, cross talk.) LAN logic **70** also includes a LAN controller **84** that, when receiving a frame from a LAN, synchronizes its internal receive clock circuitry during the seven preamble bytes, detects the LAN frame through the Start Frame Delimiter (SFD) byte, checks the data integrity of the frame by computing/comparing the four Frame Check Sequence (FCS) bytes, removes the protocol information such as preamble bytes, SFD byte and FCS bytes, and deserializes the remaining incoming bits to provide data bytes at the parallel interface with bus **72**.

When transmitting data bytes from the parallel interface with bus **72** to the LAN, LAN controller **84** serializes the incoming parallel bytes, generates the protocol information bytes, and computes and sends the FCS bytes.

In a preferred embodiment of the present invention, the LAN controller **84** is a master device with an internal direct memory access (DMA) controlling the transfer of bytes on the parallel interface with bus **72**.

Switch logic **74** includes a switch connector **86**, a SCC **88** for transmitting serial data to the switch card through connector **86** and receiving data from the switch card through connector **86**, a control logic **90** for generating the request signal and synchronizing the timing between the switch card and the LAN adapter, and a clock multiplier **92** for providing control logic **90** with the transmit clock generating the request signal at a frequency sixteen times the frequency of the data clock.

Connector **86** couples the LAN adapter to the switch card through a back plane and carries the request signal (REQ), the grant signal (GNT), the transmit data signal (DATA OUT), the receive data signal (DATA IN), the data clock signal (DATA CLK) and the synchronization signal (SYNCHRO).

When transmitting data bytes from the parallel interface to the switch card, the SCC **88** generates high-level data link control (HDLC) frames. The HDLC generation includes generating an HDLC flag (one byte) to start a frame, serializing and sending the incoming parallel data bytes, computing and sending the FCS (two bytes) after the data bytes, and generating an HDLC flag (one byte) to end the frame.

When receiving an HDLC frame from the switch card, SCC **88** detects the incoming frame through the flag, checks the data integrity of the frame by

computing/comparing the Frame Check Sequence (FCS), and deserializes the incoming bits to provide data bytes at the parallel interface.

In a preferred embodiment of the present invention, the SCC **88** is a master device with an internal DMA controlling the transfer of bytes on the parallel interface.

System bus logic **76** includes a microcontroller **94** and a memory **96**. Microcontroller **94** includes a processing unit, a read-only storage (ROS) for storing the operational code, a random-access memory (RAM) that operates as a cache memory, a programmable chip select for generating a memory chip select (CS1), a LAN controller chip select (CS2), a serial communication controller chip select (CS3), and a control logic chip select (CS4).

Memory **96** transfers the LAN frame between the LAN controller **84** and the serial communication controller. Such a memory is divided into at least two independent areas: a LAN-to-switch area organized in a first plurality of 2K bytes buffers and a switch-to-LAN area organized in a second plurality of 2K bytes buffers.

General bus **72** includes a data bus, an address bus and control signals such as read, write, chip selects, interrupts, bus requests and bus acknowledges, but the width of both data bus and address bus is not critical.

Following a machine power-on or a reset, microcontroller **94** initializes the three main components of the LAN adapter: memory **96**, LAN controller **84**, and SCC **88**, as illustrated in **Figure 5**. The initialization of LAN controller **84** includes setting up the receive DMA of the controller with the base address of the LAN-to-switch buffer no. 1 in memory **96**. The initialization of SCC **88** includes setting up the receive direct memory access (DMA) of the SCC with the base address of the switch-to-LAN buffer no. 1 in memory **96**.

Assuming that a frame is received from the network on the receive line TD of connector **60**, this frame is converted into TTL logic by analog circuitry **82** and transferred to LAN controller **84**. While the incoming bits are stored in an internal receive first-in-first-out (FIFO) queue, the receive DMA of LAN controller **84** requests the use of general bus **72** to arbiter **78** by activating the HOLD signal. When the general bus **72** is free, arbiter **78** activates the HLDA signal. The receive DMA of LAN controller **84** transfers the bytes of the frame from the FIFO of the LAN controller and they are stored into the LAN-to-switch buffer no. 1 in memory **96**. When the entire frame is stored in the memory, LAN controller **84** activates its interrupt signal INT1.

When receiving the interrupt signal INT1, microcontroller **94** stops its current task to execute a LAN interrupt routine by reading the interrupt register of LAN controller **84** to determine the cause of the interruption, initializes the receive DMA of LAN controller **84** with the base address of the LAN-to-switch buffer no. 2 in memory **76** (At this time a new frame coming from the network can be received), reads the frame byte count and the destination address, and jumps to a switch interface routine.

When running the switch interface routine, microcontroller **94** determines the address of the destination LAN adapter using routing tables (it can be a unique address, a multicast address or a broadcast address), determines the connection time by dividing the frame count by fifty-three, stores both the destination address and the connection time in a parallel-to-series register located in control logic **90**.

Referring to **Figure 6**, there are depicted the components controlled by control logic **90**. When receiving the CS4 signal from the microcontroller **94**, control logic **90** stores into a memory **102** by the intermediary of a data bus interface **100** the destination address and the connection time which are received from

microcontroller **94** via bus **72**. A signal CS4 is transmitted from microcontroller **94** to control logic **90** each time microcontroller **94** is interrupted by the LAN controller **84** when a data frame is received. Then, a finite state machine (FSM) **104**, which controls all the operations of control logic **90**, starts sending a REQ signal containing the destination address and the connection time through a switch interface control unit **106**. Note that FSM **104** is timed by a clock control unit **110**, which also provides the clock of the REQ signal transmitted by switch interface control unit **106** from the data clock multiplied by sixteen.

When receiving a GNT signal, the switch interface control block **106** is activated for storing into the memory **102** the GNT contents and interrupting the finite state machine **104**. The latter compares the destination address and the connection time of all requests stored in the memory **102** with the GNT contents. If the comparison is positive, FSM **104** programs the transmit DMA of the SCC **88** through the SCC interface **108** by sending to SCC **88** the destination address and connection time through data bus **72** utilizing data bus interface **100**. FSM **104** also activates the Clear-to-send line (CTS) to the SCC **88** through the SCC interface **108**. The transmit DMA transfers the bytes from the LAN-to-switch buffer no.1 of memory **96** into the switch card according to the received destination address and connection time of the GNT signal. These bytes are sent in an HDLC format to guarantee the data integrity through the backplane. When the LAN-to-switch no.1 is empty, SCC **88** activates its interrupt line INT2. It must be noted that the HDLC format uses a flag when the end of the frame is reached even if the last slot is less then fifty-three bytes, and does not require the use of padding bits to complete a fifty-three byte cells as in the ATM procedure.

Control logic **90** synchronizes the timing of the different actions described above, such as outputting the destination address and the connection time on the

request signal, receiving the grant signal and setting up the CTS signal, with the timing of the switch card, as illustrated in **Figure 7**.

When receiving the interrupt signal INT2 from SCC **88**, microcontroller **94** stops its current task to execute a SCC interrupt routine by reading the interrupt register of SCC **88** to determine the cause of the interruption and releasing the LAN-to-switch buffer no.1 in memory **96**.

When SCC **88** detects the reception of a frame from the switch card, SCC **88** requests the use of the general bus **72** to arbiter **78** by activating its HOLD line and stores the incoming bits in an internal receive FIFO. When the general bus is free, arbiter **58** activates a HLDA line to SCC **88**. From now on, the receive DMA of SCC **88** transfers the bytes of the frame from the FIFO of SCC **88** into the switch-to-LAN buffer no.1 in memory **96**. When the entire frame is stored in memory **96**, SCC **88** activates its interrupt line INT2.

When receiving the interrupt signal INT2, microcontroller **74** stops its current task to execute the SCC interrupt routine by reading the interrupt register of SCC **88** to determine the cause of the interruption, initializing the receive DMA of SCC **88** with the base address of the switch-to-LAN buffer no.2 of memory **96** (at this time a new frame coming from the switch card can be received), initializing the transmit DMA of LAN controller **84** with the base address of the switch-to-LAN buffer no.1 of memory **96** and the byte count, and starting the transmit DMA of LAN controller **84**.

Then, the transmit DMA of SCC **88** requests the use of general bus **72** to arbiter **78** by activating the HOLD line of SCC **88**. When the general bus is free, arbiter **78** activates the HLDA line to SCC **88**. The transmit DMA of SCC **88** transfers the bytes of the frame from switch-to-LAN buffer no.1 of memory **96** to the

LAN. These bytes are transmitted serially through analog circuitry **82** onto the transmit line TD of connector **80**. When the entire frame is sent out, LAN controller **84** activates the interrupt line INT1 to microcontroller **94**.

When receiving the interrupt signal INT1, microcontroller **94** stops its current task to execute the LAN interrupt routine by performing the actions of reading the interrupt register of LAN Controller **84** to determine the cause of the interruption and releasing the switch-to-LAN buffer no.1.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will also be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.